WG14/N1083
INCITS/J11-04-002

JTC1 SC22/WG14
Redmond Meeting Minutes (draft)
25-29 October, 2004

Meeting Times:

25 October 2004 09:30-12:00 13:30-17:00
26 October 2004 09:00-12:00 13:30-17:00
27 October 2004 09:00-12:00 13:30-17:00
28 October 2004 09:00-12:00 13:30-17:00
29 October 2004 09:00-12:00

Meeting Location:

Microsoft Corp.
1 Microsoft Way
MS Conference Center
Redmond WA, USA 98052

Meeting Host:

USA / ANSI, Microsoft

1.0 Opening activities, Monday, October 25, 2004 The meeting convened at 9:30 AM PST.

1.1 Opening Comments:

Herb Sutter, Microsoft, welcomed everybody and described the local arrangements.

1.2 Introduction of Participants/Roll Call

John Benito	WG14 Convener	USA	
Tom Plum	Plum Hall	USA	
Barry Hedquist	Perennial, Inc	USA	
John Parks	Intel	USA	
Fred Tydeman	Tydeman Consulting	USA	
P. J. Plauger	Dinkumware, Ltd	USA	
Tana L. Plauger	Dinkumware, Ltd	USA	
Randy Meyers	Silverhill Systems	USA	
Douglas Walls	Sun Microsystems	USA	HOD
Nobu Mori	SAP	Germany	HOD
Francis Glassborow	Self	UK	
Herb Sutter	Microsoft	USA	
Jeff Muller	Oracle	USA	
Martyn Lovell	Microsoft	USA	
David Keaton	self	USA	
Cecilia Galvan	Metrowerks	USA	
Nick Stoughton	FSG	Cat A Liaison SC 22	
Clark Nelson	Intel	USA	
Peter Seebach	Self	USA	

Willam Wakker	ACE	Netherlands	HOD
Olwen Morgen	BSI	UK	HOD
Raymond Mak	IBM-Canada	Canada	HOD
Dan Gohman	Cray	USA	
John Hauser	Self	USA	
Chris Walker	Dinkumware, Ltd	USA	
Anson Tsao	Microsoft	USA	
Alessandro Contenti	Microsoft	USA	

1.3 Selection of Meeting Chair

JB - meeting chair.
BH - meeting secretary.

1.4 Procedures for this Meeting

The Chair announced the procedures are as per normal. INCITS J11 members are reminded of the requirement to follow the INCITS Anti-Trust Guidelines which can be viewed at http://www.incits.org/inatrust.htm.

All 'N' document numbers in these minutes refer to JTC1 SC22/WG14 documents unless otherwise noted.

1.5 Approval of Previous Minutes (N1064)

Prior Minutes from meeting in Coogee, Australia, 29 March - 02 April, 2004 are approved as revised w/o objection. New number N1081
(Motion: Walls/Tydeman, unanimous consent)

1.6 Review of Previous Action Items and Resolutions

1. OPEN - Reassigned DR group ACTION: DOUG GWYN to develop words, RoR, DR-236, restricting use of union members.

2 DONE ACTION: Convener - Forward a proposed ISO/IEC 9899:1999 Technical Corrigenda containing corrections to the C99 Standard based on the following Closed Defect Reports: 207, 211, 215, 218, 222, 223, 224, 225, 228, 229, 230, 233, 238, 239, 240, 241, 242, 243, 244, 245, 247, 248, 249, 250, 262, 263, 265, 267, 269, 270, 272, 273, 274, 275, 276, 278, 279, 281, 282, 285.

3. OPEN ACTION: Tom Plum will communicate our discussion of Core Issue 268 to the C++ Core Group.

4. OPEN ACTION: RM to propose a RoR for DR 219.

5. OPEN ACTION: RM to edit grammar in Proposed TC for DR 289.

1.7 Approval of Agenda (N1073)

New agenda items:

12. scope - Plum Wed

13. helpers for malloc - Wed

Agenda approved as modified (Motion: Tydeman / Walls, unanimous consent)

1.8 Distribution of New Documents

None

1.9 Information on Next Meeting (N1070)

Next meeting to be held in Lillehammer, Norway, Apr 4-8, 2005. Hotel information, etc., provided in N1070. Get reservations by EOY. WG21/C++ will follow the C meeting, April 10-15, 2005. Note that room reservations MUST be made through the host.

Oct 2005, hosted by Canada, dates are firmed up, but... possible switch of dates w/C++, Mont Trembant, Quebec

1.10 Identification of National Bodies (Benito)

Countries represented: UK, Canada, US, Germany, Netherlands.

1.11 Identification of J11 voting members (Tydeman)

12 voting J11 members out of 15 possible members. (See attached J11 membership list for attendees.)

2. Liaison Activities

2.1 INCITS/J11 (Walls, Meyers)

A delay in getting a NWI ballot for the special math functions is getting resolved. Ballot closes 15 Dec.

INCITS/J3 asked that we review a new header file for FORTRAN header types. OM volunteered to review, ask DG to review as well. FT pointed out that this new header file applies to FORTRAN 2003 std.

2.2 SC22/WG11 (Wakker)

Complex function document in ballot.

2.3 SC22/WG14 (Benito)

TC2 submitted with 75 corrections, will incorporate TC2 and TC1 as a revision to C99 for the next publication (C04, or C05).

The new TRs should be listed as 'free'. TC2 ballot complete, at ITTF for publication. Will ask for a revision to 9899 consisting of C99 + TC1 + TC2, and nothing else.

NWI for special math in ballot at SC22. Secure C DTR has an official number - 24731.

2.4 J16/WG21 (Sutter)

WG21 met last week. Library extension TR completed, which includes the special math functions. Evolution working group splitting into several smaller groups: generic programming; security/performance; libraries/modules; ease of use thru library extensions or core changes as needed. The work of these groups is intended to result in normative changes / additions to the C++ Standard.

2.5 FSG - Free Standards Group (N1082) (Stoughton)

FSG submitted a draft of LSB doc to ITTF, includes a number of extensions to C and POSIX libraries. No C++ interfaces included. Tracking work on security TR, some overlap with work in Austin Group, parallel functionality. SC22 established a POSIX advisory group that will liaise with AG (?). RM identified some security functions that were similar to or the same as UNIX functions, and we made some name changes to those functions - essentially to go our own way. PJ pointed out there was good rationale for the names chosen, whether the same, or similar, and none of the changes were gratuitous.

2.6 Other Liaison Activities

SC22 plenary in Korea, WG15 and WG20 were disbanded. An SC22 RG for I18N formed, as well as an SC22 Advisory Group for POSIX. NS asked if WG14 would like to establish a formal liaison with the SC22 Advisory Group for POSIX.

There is a mirror to the WG14 web site, wg14.dinkumware.com.

3. TR Status Report (TR 18037: C Extensions for Embedded Processors, N1021, N1071) (Wakker)

ITTF requested changes to the document title, and clause numbering to match ISO directives. Implementations are underway. N1071 identifies issues found to date, for possible revision or TC to TR. Specific items will be discussed in the DR portion.

General plan is to generate a list of issues / defects, that can then be processed as a TC to the TR.

12 issues listed in N1071 for discussion.

Issue 1: Name conflict for strtok - The name strtok is already a standard C function. Need a new name. Agreed to strtofx()

Issue 2: Order in which overflow handling and rounding is done. Present spec of handling overflow prior to rounding conflicts with floating point. Suggestion to make it implementation-defined. Agreed to reverse order to match FP.

Issue 3: Typo in 4.1.6.2.1 - Binary arithmetic operations - change wording to "integer type result"

Issue 4: Type generic macro's - spec is either incomplete or wrong. fx is supposed to be part of the name, Agreed that 'fx' is intended to be part of the name.

Issue 5: Typo in new text for 6.2.6.3 - replace 'integer' types with 'fixed-point' types

Issue 6: fp arithmetic support functions do not specify what happens if an integer result overflows - what happens if an integer result overflows? Note to make it undefined.

Issue 7: Error in 7.18.a.6.3 fractional bits should not be set to zero when saturation has occurred. Agreed

Issue 8: Diagnostic required on named-register constraint violation? Violates a 'shall' . JH needs to review this. Decided this is a problem, remove the existing constraint, write a new one the compiler can check.

Issue 9: Effective type definition (aka US-40) - issue hinges on the presence of address-space qualifiers. Proposed a new definition. Problem is not clear. Propose to expand the definition of effective type by striping away the address qualifier.

Issue 10: The relationship between named-registers and external object definitions. Relationship needs to be specified. Possible interaction with #8 above. Decided this is not an issue at all, no change.

Issue 11: Initialization of global named-registers - initialization mechanism is not specified. JH wants to take some time to look at this - OPEN

Issue 12: Address space qualifier in specifier-qualifier-list - error in definition: Proposed to change constraint to "Within a structure or union specifier, the type of a member shall not be qualified by an address space qualifier." OK

Two additional items added, #13, #14 - will be included in action item.

ACTION: WW to generate an issues list with proposed resolutions.

4. TR Status Report (TR 19769: C Extentions for New Character Types, N1040) (Mori)

TR is approved and submitted to ITTF, no comments from ITTF yet.

5. Defect report status (Benito)

TC2 was submitted with the following DRs: 207, 211, 215, 218, 222, 223, 224, 225, 228, 229, 230, 233, 238, 239, 240, 241, 242, 243, 244, 245, 247, 248, 249, 250, 262, 263, 265, 267, 269, 270, 272, 273, 274, 275, 276, 278, 279, 281, 282, 285.

DRs currently in REVIEW status: 251, 261, 266, 268, 284, 287, 289, 290, 292

DR298 affects C++, status updated in mailing.

DR291, typo. Reference to J.3.1.2 should be J.3.12.
ACTION Convenor - correct the above typo in DR 291.

6. Special Math Functions (N1076) (Plauger)

N1076 points out issues with calculating results from certain Bessel functions, and proposes an upper limit of 128 on the function parameters.

TR-1 from C++ has been voted out for submittal. Believes that the functions have been defined in a way that enables them to be computed accurately. Assembling a bibliography to provide information on how to compute these functions. Expect to have more concrete words in a couple of months. SC22 is the sponsor of the NWI, and currently in SC22 ballot.

7. C++ Preprocessor Changes (N1068) (Nelson)

N1068 addresses synchronization of the preprocessor between C and C++. The C++ version was taken from C89, and has changed a bit. The technical content of this paper has been accepted by WG21, most changes are editorial. Two synchronization item are acceptance of numbers and underscores in a preprocessor name. Accepting these as "defects" is the quickest way to proceed with these - there are nine items total.

RM not completely comfortable with using DRs as a mechanism to back door a change to the standard. PJ agreed, but believes in this case the 'defect' is simply to correct synchronization with C++ for an item that should stay in sync. Discussion on whether this should be split into nine DRs, or handled as one. General consensus that the items should be handled as a defect.

Item #1, clarification: a non-digit is only the letters a-z, upper and lower case, and underscore _. This is the only item that is a 'technical' change.

Item #2: C99 changed the definition of 'preprocessing directive'. Neither C89, nor C++ do this, so some wordsmithing is needed to preserve the intent.

Item#6: Needs wordsmithing to simplify / clarify the meaning.

Assigned DRs 302 - 310. Submitted from WG21.

ACTION: Clark Nelson to convert N1068 paper into a set of DRs

8. C Extensions for Decimal Floating-Point Arithmetic (DTR 24732) (N1077) (Kwok)

N1077 is working draft 4 of the DTR. Fred T has provided some comments via email, but otherwise very few comments have been submitted. The TR needs to be ready for Registration by April 2006. Issues of how to migrate to DFP from another representation, i.e. BFP, are not yet resolved. Current proposal calls for three new types, and we are not ready to make a decision on that approach. There is also some concern that programs will try to make use of both BFP, and DFP according to specific needs. The proposal identifies some rules for dealing with going from one to the other that need, but the safety in doing so is questioned. Francis pointed out that the Standard does not "require" BFP.

Should implicit conversions between DFP and BFP be allowed? Discussion ranged from banning them completely to keeping them limited (as few as possible). Issue needs to be worked and will be presented as two options.

How to deal with const values? An approach will be added for further discussion.
How to handle literals?
How will this document be synchronized with IEEE 754 work? JB stated we want to stay 'behind' their progress, rather than accelerate past them.

C++ is leaning toward an approach using class libraries.

9. C Extensions for Library Security (DTR 24731-N1078, N1079, N1080) (Meyers)

N1078 - DTR - C Library
N1079 - TR Editor's Report
N1080 - Limited size_t

Randy Meyers reviewed the latest version of the C Library Security TR per the Editors Report.

9.1 - N1078, N1079

N1078 - 4.1.1;p4 - do not want to introduce undefined behavior in this library - make it a shall, issue a diagnostic. Some discussion of whether or not to force termination of compilation at that point. No. Doing so is considered to be outside the scope of the Standard.

Returns of error codes, zero for success, non-zero for non-success

N1079 - scanf - Considerable discussion on implementation of the scanf conversion specifier that was decided on in Sydney. Should the default precision for the new _s scanf family really be .0, or should it be .*? The .* is more consistent with previous drafts, and means that format strings need not be edited when converting from fprintf to fprintf_s, as long as the array sizes are provided in the parameter lists.

N1078 - General. Why do instances of "undefined-behavior" exist in this TR? This area needs to be investigated further, to tighten these up, avoid writing outside destination arrays, etc. General consensus to minimize undefined-behavior where possible.

N1078 - 4.6.1.1 - Discussion on the sample implementation code for asctime.
Straw Poll, asctime
Remove sample implementation - LOTS
Remove requirement to call mktime - LOTS

9.2 - N1080 - limited size_t

Should the 'size_t' to secure functions be limited? One approach is to create a new typedef, size_t_s, that can be range checked. Headers containing size_t would also contain "typedef size_t size_t_s", and be protected by the secure library macro __STDC_WANT_SECURE_LIB__. The type would have upper and lower checkable limits. In discussing the lower limit. TP pointed out that the C Standard requires 16 bit registers, and anything smaller than that does not conform to the C Standard. NS: POSIX uses a type called ssize_t that eliminates all negative values, and solves much of the problem being addressed here. ML believes adding the type has value, even if nothing can be said about the upper limit. PJ: Arguments of this type should be checked to insure they are smaller than the largest addressable value the system provides - there is always some size determined by the implementation that can be checked. Allow the implementation to pick the limit.

Note: N1080 pg4: "bool issue.... SIZE_MAX;" should be "bool issue....SIZE_MAX_S;"

Should the limit size_t_s be defined somewhere, as in SIZE_MAX_S ?

Straw Polls:

A. Should we pursue some form of bounds checking for size_t?
Yes - LOTS, No - none

B. Can the implementation be permitted to place the limit smaller than some very large objects?
Yes-LOTS, No-2

C. Range of reasonable values?
- implementation defined, with a recommended practice section

Yes-LOTS

- use POSIX approach, all negative numbers are invalid - still allowed with above

D. Should there be an object-like macro (e.g. SIZE_MAX_S) that has type compatible with size_t, whose value is an rvalue expression? The name is only a place holder.

Yes - 11, No - 7 , Abstain - 3

Example:

```
if(x <= SIZE_MAX_S) {...}
...
```

E. Do we want a typedef for a limited size_t?

yes - 12, no - 5, abstain - 4

F. What is an acceptable name:

	yes	no
size_s_t	13	5
size_t_s	10	8
bsize_t	16	2

bsize_t has lots of hits on Google - gcc - dropped

More discussion on item F - added:

osize_t	10/3
rsize_t	16/2 <= consensus choice, no hits on Google

9.3 scanf issue - revisited (See 9.1 above)

Randy wants to revisit how the scanf functions handle their conversion / format specifiers, PJ believes we are to far down the road to make the change now

Straw Poll

A. In favor of not using precision to specify the length of strings for scanf_s?

-- No differences from C99 scanf

-- scanf_s just always takes an extra argument after the char*

yes - 13 no - 0, abstain - 8

NS raised an issue of every pointer being a pair.

B. Do we want to pass a size following every pointer following fmt in scanf_s?

No consensus to take a vote on this.

9.4 new _s functions - Martyn

Martyn Lovell presented a list of new '_s' functions to be added to the TR.

File Set:

- tmpfile_s
- fopen_s
- freopen_s

Multibyte/Wide Character Set

- mbstowcs_s
- wcstombs_s
- wctomb_s

- mbstrowcs_s
- wcrtomb_s
- wcrstombs_s

Printf Set
- printf_s family

Sprintf Set
- sprintf_s family

Truncate Concepts

ML also proposed that %n be removed from the printf_s family new proposed functions. They have disabled %n in the printf family, but have a way to turn it back on for conformance purposes. The use of %n is inherently dangerous since it enables a system to retain the user input, and pass it on in a multistage attack, i.e. confidential user information can be captured and passed on to an unauthorized third party.

Straw Polls
A. Straw Vote: Proceed with above File Set:
yes-14 no-1 abstain-6

B. Straw Vote: Proceed with above Multibyte/Wide Character Set
yes-16 no-0 abstain-6

C. Straw Vote: Proceed with above printf_s family and changes:
yes-14 no-0 abstain-8

D. Straw Vote: Proceed with above sprintf_s family and changes:
yes-15 no-0 abstain-7

E. Straw Vote: Develop an editor's note above TRUNCATE concept:
yes-14 no-1 abstain-7

9.5 Discussion on 4.4.1.1, rand_s. General consensus that there is no way to really do 'secure' random number generation, so rand_s() should be dropped from the TR.

ACTION: Nick S to provide a liaison copy of a revised version of the Security TR to the Austin Group meeting in January, 2005.

10. DEFECT REPORTS (Vers 1.16 summary)

10.1 Potential Defect Reports (N1072, N1074, N1075, N1084 (Tydeman)

10.1.1 N1072 (Defect Report #299) Summary: The standard is not clear as to which type-generic macro(s) should be used to compute the absolute value of real, complex, and imaginary types. cabs() is not a generic name. This item should be a DR. Assigned DR 299.

10.1.2 N1074 (Defect Report #300) Summary: The standard does not require translation-time expression evaluation to produce "obvious" values. For example, there is no requirement that static const double d = 1.23/4.56 - 1.23/4.56; be zero. Agreed to make this DR 300.

10.1.3 N1075 (Defect Report #300) Summary: Exactly WHERE are the MEANINGS of any of the FE_* macros defined in cases where <fenv.h> applies to an environment that is not IEEE-754 (IEC 60559)? Assigned DR 301.

10.1.4 N1084 (Defect Report #311) Summary When FLT_EVAL_METHOD is 2, the 0.1f is represented to the precision of long double, while the type remains as float. Then, when the cast to double is done, contradictory requirements are specified by the standard. One part of the standard requires that when a float is promoted to a double, the value is unchanged. While another part of the standard requires that extra precision be removed by the cast conversion. Assigned DR311.

10.2 Existing Defect Reports

DR259, DR259 Discussion - This DR is being reviewed as a result of issues raised on the news group (comp.std.c). Proposes that wording from the rationale be added to the standard: "A function-like macro invocation f() has the form of either a call with no arguments or a call with one empty argument. Which form it actually takes is determined by the definition of f, which indicates the expected number of arguments." General consensus that "the Standard is clear", as originally answered, and as noted, the text is already in the Rationale.

DR296 DR296_details Proposed response does not fully capture the intent - needs additional words. Fred T provided additional details of all functions that have an infinity as an input, or infinity as an output, as specified in Annex F.
ACTION: Convenor to add the text from FT to the Committee Discussion portion to DR 296.

DR219 and DR236 have both been open for some time.
ACTION Olwen Morgen to have a paper to close out DR219 in April 2005

DRs in REVIEW Status

DR260 - Review status. Moved to CLOSED - RR

DR283 - Review Status. Moved to CLOSED - TC

DR286 - Review Status. Moved to CLOSED - TC

DR288 - Review Status. Moved to CLOSED - RR

DR289 - Review Status. Moved to CLOSED - TC

DR291 - Review Status. Moved to CLOSED - TC

DR293 - Review Status. Moved to CLOSED - TC

DR294 - Review Status. Moved to CLOSED - RR
ACTION: Rationale Editor to add words to rationale for DR294.

DR295 - Review Status. Moved to CLOSED - TC

DR297 - Review Status. Moved to CLOSED - TC

DRs in OPEN Status:

DR296 - OPEN. Was never the intention to allow an overflow for the case cited - Standard is in error Move suggested TC to Proposed TC, add proposed rationale to Proposed Committee

Response, add Fred T email to the Committee Discussion. Moved to REVIEW.

DR299 - The standard is not clear as to which type-generic macro(s) should be used to compute the absolute value of real, complex, and imaginary types. Move Suggested TC to Proposed TC. Moved to REVIEW.

DR300 - The standard does not require translation-time expression evaluation to produce "obvious" values. For example, there is no requirement that static const double d = 1.23/4.56 - 1.23/4.56; be zero. Discussion as to whether or not this constitutes a new requirement in the Standard. TP prefers that we address the issue of FP precision / accuracy across the board, rather than piecemeal. FP accuracy is of concern to the High Integrity community to the extent that FP is avoided. Agreed that this is not a defect, and that this DR imposes a new requirement on implementation. Proposed words: "This is not really a requirement, but an area to be investigated that could be addressed in a future revision of the C Standard." Agreed to.

DR301 - Exactly where are the meanings of any of the FE_* macros defined in cases where <fenv.h> applies to an environment that is not IEEE-754 (IEC 60559)? Considerable discussion on the need for these proposed changes, as well as other FP issues in general, to address specific needs of specific communities. PJ pointed out that the standard's macros lack semantics, and the proposed changes say too much, and not enough. Yes these items need to be addressed, but ..
Straw Poll : 1. Should we make a TC along the lines outlined in DR301? Yes - 4, No - 10, Abstain - 5.

10.3 NEW DRs

N1068 (Changes from the C++ Preprocessor Description) (Nelson) - C Preprocessor Items to synchronize with C++ - to be turned into 9 DRs, 302-310. Treat all of these as OPEN. The proposed changes are basically redundant, resulting in discussion about why redundant changes should be made at all. Reviewed each item in N1068:

Item #1. likely TC item - wording change Straw poll - support as written Yes - 17, No - 0, Abstain - 2.

Item #2. FT asked that the DR reference be included.

Item #3. RM - is this the right paragraph (6.10.1) - JP - yes. Does the constraint also appear in C++? The proposed changes are basically redundant, resulting in discussion about why redundant changes should be made at all. Straw poll to adopt constraint as written? Yes-4, No-13, Abstain-3. Clark will tell C++ the words are redundant and should be removed.

Item #.4. These words are redundant. Straw Poll: Adopt words in paper? yes - 3, no - 8, abstain - 7
Straw poll: adopt words "..all remaining identifiers, including those lexically identical to keywords, are replaced with the pp-number 0.", yes-lots.

Item #5. Correct Para number. Straw poll: words as presented?: yes- 13, no- 4, abstain- 1
Item #6. Poll: accept words: "..defines a function-like macro with parameters, whose use is similar syntactically to a function call." Yes - lots

Item #7. accept as written.

Item #8. accept as written

Item #9. accept as written.

Two additional, related, items:

Item #10, C99: para 6.10.8;p2 - C++ Core asked: "Under what conditions?" Already defined. No further clarification required.

Item #11. C99: para 6.10.3;p12 - C++ Core asked: Is "..are merged.." correct English? - Yes. No further action.

11. WG14 Administration (Benito) - See #15

12. Scope (Plum) (N1062)

This paper addresses C language synchronization of a proposal presented in C++ (WG21 N1614) titled "#scope: A simple scoping mechanism for the C/C++ preprocessor.", which is being considered in the WG21 C++ Evolution Working Group. Proposals considered in that group are intended to be future normative changes to the C++ Standard.

Three classes of problems to be solved:
1. NameHijacking - the unintended replacement of ordinary identifiers by macro names
2. NameClashes - the unintended use of the same macro names in different headers for different purposes
3. PrefixSchemes - name-prefixing schemes which can reduce, but never eliminate, name clashes

From the C++ meeting last week, a hybrid solution has been developed, which was extracted from the WG21 Wiki:

1. Solve the "#import name-clash" problem (see AvoidingPriorKeywords) by adding an "s" to the keywords, i.e.
#imports A, B, C
#exports X, Y, Z

2. The syntax and "begin-end" matching of the "new region" marker remains the most contentious issue; see below. For purposes of discussion, let's use an obviously-invalid place-holder:
xxx-begin-macro-region
  ...
xxx-end-macro-region

3. Whatever we use for "begin-end" region, the #imports and #exports can be hidden from "old" preprocessors using the usual feature-test approach. The semantics of #imports and #exports remain as originally proposed.
#ifdef __std_macro_imports_exports // or whatever
#imports A, B, C
#exports X, Y
#exports Z
#endif // __std_macro_imports_exports
 ...

4. A preprocessor that honors #imports and #exports is a "new" preprocessor. A "new" preprocessor causes the "local" macro names in each "macro region" to be "not visible" outside the region, and the not-imported names from outside do not conflict with the "local" macro

names, as per Bjarne's original paper.

5. Obviously, a library vendor targeting multiple platforms must use pp syntax which can be compiled with today's preprocessors, until all their target environments implement the new pp syntax. The feature-test described above (e.g. __std_macro_imports_exports) solves part of this problem.

6. Along with the "new region" marker (whatever that might be), today's library vendor can add some kind of "prefix()" marker, listing the name-prefixes used by "local" macro names in today's headers.

7. If a header file specifies one or more "prefix" strings, then its names must also meet the "good citizen" property; i.e., each name that is #defined in that header file must be in one of these two sets:
a. Names explicitly listed on "exports" lines;
b. Names prefixed with a prefix listed in a "prefix()" list;
Also each name that is used in that header file must be in one of these three sets:
a. Names explicitly listed on "imports" lines (including the standard-library names described in SpecialMacros);
b. Names explicitly listed on "exports" lines;
c. Names prefixed with a prefix listed in a "prefix()" list.
Also each name that is used in that header file must be in one of these three sets:
a. Names explicitly listed on "imports" lines (including the standard-library names described in SpecialMacros);
b. Names explicitly listed on "exports" lines;
c. Names prefixed with a prefix listed in a "prefix()" list.
Enforcing this list is a requirement upon every "new" preprocessor.

8. The constraints described at InteractionWithIf, NestingBehavior, and LibraryVendors all influence the "new" syntax for "macro region", as well as the work-around for "old" preprocessors. In other words, what are the "new" and "old" syntaxes for ...
xxx-begin-macro-region
  ...
xxx-end-macro-region
One proposal piggy-backs the "old" syntax onto a #if line ...
```
#if __std_macro_region(prefix) // "old" way to say  #macro_region
      // must be "true" for both old and new
 ...
#endif // end_macro_region
```

Considerable discussion on alternative mechanisms, implementation, and usability considerations.
Tom will consider the ideas presented in the discussion for inclusion / modifications to his paper.
This proposal could result in normative changes to the C Standard.

13. Helpers for Malloc (Galvan) (N1085)

Discussion of this paper was deferred to further development. Those reviewing this paper should keep in mind that it is incomplete.

14. [not used]

15. Administration

15.1 Future Meetings

15.1.1 Future Meeting Schedule

2005 Apr 4-8, Lillehammer, Norway, Oslo

2005 Oct, hosted by Canada, Mont Trembant, Quebec C current dates are 26-29 Sep, 2005, 30 Sept - 3 Oct SC22

2006 Mar/Apr - Berlin Germany 20 Mar, 27 Mar, 3 Apr, all possible week starts

2006 Fall - West Coast - TBD

15.1.2 Future Agenda Items

None

15.1.3 Future Mailings

Post Redmond meeting mailing items to be to JB by 29 Nov, 2004

Pre Lillehammer mailing items to be to JB by 7 March, 2005

15.2 Resolutions / Votes

None

15.2.1 Review of Decisions Reached

No formal decisions reached.

15.2.2 Formal Vote on Resolutions

None.

15.2.3 Review of Action Items

1. ACTION - FT to address issue of errors in Appendix G for complex multiply, and divide.

2. ACTION: Convenor to contact DG and try to get a resolution of DR236.

3. ACTION: Tom Plum will communicate our discussion of Core Issue 268 to the C++ Core Group.

4. ACTION: RM to propose a RoR for DR 219.

5. ACTION: RM to edit grammar in Proposed TC for DR 289. - DONE

6. ACTION: Clark Nelson to convert N1068 paper into a set of DRs (302-310)

7. ACTION: Convenor to add the text from Fred T email to the Committee Discussion portion in DR 296.

8. ACTION Olwen Morgen to have a paper to close out DR219 in April 2005

9. ACTION: Rationale Editor to add words to rationale for DR294.

10. ACTION: Nick S to provide copy of Security TR to AG as Liaison

11. ACTION Convenor - correct the typo in DR 291: Reference to J.3.1.2 should be J.3.12.

12. ACTION: WW to generate an issues list with proposed resolutions for Embedded TR

13. ACTION RM, deliver draft revision of Security TR to review team: Plum, Hedquist, Stoughton, Keaton, Galvan, Seebach.

13 ACTION: Convenor - Forward Revised draft of Security TR to SC22 for Registration upon approval of the Editorial Committee.

15.2.4 Thanks to Host - Thank you Microsoft!!

Thanks to Dinkumware for the network support.

15.3 Other Business

None.

16. Adjournment at 3:47 PM, Thursday, 28 October, 2004
(Motion Walls/Tydeman) unanimous consent

==========

J11/ WG14 US TAG Meeting - Wed, 27 Oct 2004

Convened at 4:29 PM

Attendees:

John Benito	WG14 Convener	USA	
Tom Plum	Plum Hall	USA	
Barry Hedquist	Perennial, Inc	USA	
John Parks	Intel	USA	
Fred Tydeman	Tydeman Consulting	USA	
P. J. Plauger	Dinkumware, Ltd	USA	
Tana L. Plauger	Dinkumware, Ltd	USA	
Randy Meyers	Silverhill Systems	USA	
Douglas Walls	Sun Microsystems	USA	HOD
Jeff Muller	Oracle USA		
Martyn Lovell	Microsoft	USA	
David Keaton	self	USA	
Cecilia Galvan	Metrowerks	USA	
Nick Stoughton	FSG	Cat A Liaison SC 22	
Clark Nelson	Intel	USA	
Peter Seebach		Self	
John Hauser	Self	USA	
Chris Walker	Dinkumware, Ltd	USA	
Anson Tsao	Microsoft	USA	
Alessandro Contenti	Microsoft	USA	

15 voting members, 12 voting members present

1. US position on Special Math functions NWI (N1056)

J11 recommends approval of ISO/IEC JTC 1/SC 22 N3814 - New Work Item Proposal for a Technical Report Type 2 on Extensions for the Standard Library of the Programming Language C to Support Mathematical Special Functions with the following recommendation to the six questions:

New Work Item Proposal for a Technical Report Type 2 on Extensions for the Standard Library of the Programming Language C to Support Mathematical Special Functions

Q.1 Do you accept the proposal in document SC 22 N3729 as a sufficient definition of the new work item? (If you have responded "NO" to the above question, you are required to comment.)

__X_Yes
____No

Comments:

Q.2 Do you support the addition of the new work item to the SC 22 programme of work?
__X_Yes
____No

Comments:

Q.3 Do you commit yourself to participate in the development of this new work item?
__X_Yes
____No

Comments:

Q.4 Are you able to offer a project editor who will dedicate his/her efforts to the advancement and maintenance of this project? (If "YES," please identify)
__X_Yes
____No

Comments:

P.J. Plauger
Dinkumware, Ltd.
398 Main Street
Concord MA 01742
pjp@dinkumware.com
+1-978-371-2773

Q.5 Do you have a major contribution or a reference document ready for submittal?

__X_Yes
____No

Comments:

WG14/N1051 Technical Report on Special Math functions

Q.6 Will you have such a contribution in ninety days?
_X__Yes
____No

Comments:
---- end of motion -----

(Seebach, Hedquist) Passed (11,1,0,15)

2. Appoint delegation and HOD for future WG14 meetings.

Douglas Walls (HOD), John Parks, Doug Keaton
PASSES - (12,0,0,15)

3. INCITS official designated member/alternate information.
Be sure to let INCITS know if designated member or alternate changes, or if their email address changes

4. Motion to restore voting rights to US Army member, Douglas Gwinn (Walls, Seebach) (12,0,0,0)

5. Adjournment at 4:46 PM

Motion to adjourn (Plauger, Hedquist) - unanimous consent.